Amendments to the Claims

Claim 1 (currently amended): A computer program product for serializing data retrievals and updates, the computer program product embodied on one or more computer-readable media and comprising:

computer-readable program code means for creating two identical indexes, each representing an initial state for accessing stored data and each indexing a single copy of data being indexed;

computer-readable program code means for performing searches against the data using a first of the two indexes;

computer-readable program code means for performing a first update against the data using a second of the two indexes, yielding a revised index;

computer-readable program code means for serializing information on how the first update affected the second index, including how the second index was traversed for making the first update and how the second index was modified in the first update;

computer-readable program code means for switching the first index and the revised index, responsive to operation of the computer-readable program code means for performing the first update, such that the first index becomes the second index and the revised index becomes the first index;

computer-readable program code means for applying, after operation of the computer-readable program code means for switching, the serialized information to the second index, using the information about how the second index was traversed and modified to efficiently traverse and modify the newly-switched second index, thereby yielding a second index that is synchronized with, and structurally identical to, the first index; and

computer-readable program code means for performing subsequent searches against the data using the first index.

Claim 2 (previously presented): The computer program product according to Claim 1, further comprising:

computer-readable program code means for obtaining an exclusive lock on the second index prior to operation of the computer-readable program code means for performing the first update; and

computer-readable program code means for releasing the exclusive lock after operation of the computer-readable program code means for applying the serialized information.

Claim 3 (previously presented): The computer program product according to Claim 1, wherein atomic instructions are used to maintain proper synchronization between the first index and the second index.

Claim 4 (previously presented): The computer program product according to Claim 1, wherein the computer-readable program code means for serializing information further comprises computer-readable program code means for queuing a transaction, and wherein the computer-readable program code means for applying the serialized information further comprises computer-readable program code means for applying the queued transaction to the second index.

Claim 5 (currently amended): The computer program product according to Claim 1, further comprising computer-readable program code means for performing a subsequent update against the data using the second index that results from operation of the computer-readable program code means for applying the serialized information; and wherein operation of the computer-readable program code means for performing the subsequent update causes another operation of the computer-readable program code means for serializing, the computer-readable program code means for switching, and the computer-readable program code means for applying.

Claim 6 (currently amended): A system for serializing data retrievals and updates in a computing environment, comprising:

means for creating two identical indexes, each representing an initial state for accessing stored data and each indexing a single copy of data being indexed;

means for performing searches against the data using a first of the two indexes;

means for performing a first update against the data using a second of the two indexes, yielding a revised index;

means for serializing information on how the first update affected the second index, including how the second index was traversed for making the first update and how the second index was modified in the first update;

means for switching the first index and the revised index, responsive to operation of the means for performing the first update, such that the first index becomes the second index and the revised index becomes the first index;

means for applying, after operation of the means for switching, the serialized information to the second index, using the information about how the second index was traversed and modified to efficiently traverse and modify the newly-switched second index, thereby yielding a second index that is synchronized with, and structurally identical to, the first index; and

means for performing subsequent searches against the data using the first index.

Claim 7 (previously presented): The system according to Claim 6, further comprising:

means for obtaining an exclusive lock on the second index prior to operation of the means for performing the first update; and

means for releasing the exclusive lock after operation of the means for applying the serialized information.

Claim 8 (previously presented): The system according to Claim 6, wherein atomic instructions are used to maintain proper synchronization between the first index and the second index.

Claim 9 (previously presented): The system according to Claim 6, wherein the means for serializing information further comprises means for queuing a transaction, and wherein the means for applying the serialized information further comprises means for applying the queued transaction to the second index.

Claim 10 (currently amended): The system according to Claim 6, further comprising means for performing a subsequent update against the data using the second index that results from operation of the means for applying the serialized information; and wherein operation of the means for performing the subsequent update causes another operation of the means for serializing, the means for switching, and the means for applying.

Claim 11 (currently amended): A method for serializing data retrievals and updates in a computing environment, comprising steps of:

creating two identical indexes, each representing an initial state for accessing stored data and each indexing a single copy of data being indexed;

performing searches against the data using a first of the two indexes;

performing a first update against the data using a second of the two indexes, yielding a revised index;

serializing information on how the first update affected the second index, including how the second index was traversed for making the first update and how the second index was modified in the first update;

switching the first index and the revised index, responsive to performing the first update, such that the first index becomes the second index and the revised index becomes the first index;

applying, after the switching step, the serialized information to the second index, using the information about how the second index was traversed and modified to efficiently traverse and modify the newly-switched second index, thereby yielding a second index that is synchronized with, and structurally identical to, the first index; and

performing subsequent searches against the data using the first index.

Claim 12 (previously presented): The method according to Claim 11, further comprising steps of:

obtaining an exclusive lock on the second index prior to performing the first update; and

releasing the exclusive lock after applying the serialized information.

Claim 13 (previously presented): The method according to Claim 11, wherein atomic instructions are used to maintain proper synchronization between the first index and the second index.

Claim 14 (previously presented): The method according to Claim 11, wherein the step of serializing information further comprises the step of queuing a transaction, and wherein the step of applying the serialized information further comprises the step of applying the queued transaction to the second index.

Claim 15 (currently amended): The method according to Claim 11, further comprising the step of performing a subsequent update against the data using the second index that results from applying the serialized information; and wherein the step of performing the subsequent update causes repeating the serializing, switching, and applying steps.

Claim 16 (currently amended): A method of serializing access to data in a computing system, comprising steps of:

maintaining two trees as indexes to the data, a first of which is used for searches and a second of which is used for update operations, each tree having a use count associated therewith;

carrying out searches using the search tree, further comprising the steps of:

determining, for each new search request, which of the trees is currently the search tree;

incrementing the use count for the search tree;

performing the new search request using the search tree; and

decrementing the use count for the search tree, responsive to completion of the performing step; and

carrying out each update using the update tree, further comprising the steps of:

determining which of the trees is currently the update tree;

performing an update against the update tree;

serializing~~, for each update operation~~, a record of how the update ~~operation~~ affected the ~~second~~ update tree;

switching the ~~two trees~~ update tree to become the search tree and the search tree to become the update tree, responsive to completion of the steps of performing ~~each~~ the update ~~operation~~ and serializing the record; and

applying the serialized record to the newly-switched ~~second~~ update tree, provided that the use count for the newly-switched update tree has reached a value that indicates that no search requests are currently being performed against this newly-switched update tree, delaying the step of applying the serialized record if necessary until the use count for the newly-switched update tree has reached this value, and wherein the step of applying the serialized record ensures ~~such~~ that both the ~~first~~ search tree and the ~~second~~ update tree reflect ~~the~~ each update ~~operation~~.

Claim 17 (previously presented): A method of serializing access to data in a computing system, comprising steps of:

maintaining two indexes to the data, a first of which is used for searches and a second of which is used for update operations;

serializing, for each update operation, a record of how the update operation affected the second index;

switching the two indexes, responsive to performing each update operation; and

applying the serialized record to the newly-switched second index, such that both the first index and the second index reflect the update operation.

Claim 18 (previously presented): The method according to Claim 17, wherein the two indexes are B-trees.

Claim 19 (previously presented): The computer program product according to Claim 1, wherein the indexes are implemented as trees.

Claim 20 (previously presented): The computer program product according to Claim 1, wherein the indexes are implemented as hash tables.

Claim 21 (previously presented): The system according to Claim 6, wherein the indexes are implemented as trees.

Claim 22 (previously presented): The system according to Claim 6, wherein the indexes are implemented as hash tables.

Claim 23 (previously presented): The method according to Claim 11, wherein the indexes are implemented as trees.

Claim 24 (previously presented): The method according to Claim 11, wherein the indexes are implemented as hash tables.